

Final Technical Report
on
Video Retrieval Based on Language and Image Analysis



ADVANCE, Incorporated
15120 Enterprise Court, Suite 300
Chantilly, VA 20151
(703) 968-2900

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 28 May 1999	3. REPORT TYPE AND DATES COVERED Final Technical Report (5/1/97 - 5/28/99)		
4. TITLE AND SUBTITLE Video Retrieval Based on Language and Image Analysis		5. FUNDING NUMBERS MDA972-97-C-0014		
6. AUTHORS Yiqing Liang, Ph.D.				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) ADVANCE, Incorporated 15120 Enterprise Court, Suite 300 Chantilly, VA 20151		8. PERFORMING ORGANIZATION REPORT NUMBER 99-FTR-01		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Projects Agency 3701 N. Fairfax Drive Arlington, VA 22203-1714		10. SPONSORING/MONITORING AGENCY REPORT NUMBER N/A		
11. SUPPLEMENTARY NOTES		19990609 022		
12a. DISTRIBUTION/AVAILABILITY STATEMENT UNRESTRICTED		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) This report summarizes the technical work ADVANCE Inc. performed under DARPA's contract no. MDA972-97-C-0014. This project has been conducted during an approximate two and one half year period in Phase I and Phase II implementation. The work performed has been focusing on the research and development of technologies for the indexing and retrieval of digital video content. The methodologies are built upon the theories of frame-based access and object-access, which allows users to index and retrieve video contents accordingly. Research is also performed on access at levels of video scene and video programs. For these purposes, algorithms for video shot boundary detection and keyframe extraction were developed. Closed captioning is extracted and associated with video shots so that keyword searching can be performed to retrieve video shots. Different approaches are used to segment video objects, including motion-based segmentation, and the combination of motion information and image cues for segmentation. Special objects such as human face and military target of interest were investigated as indexing points. Techniques for object identification and tracking were also studied for indexing and retrieving video contents. A comprehensive prototype has been built and software has been developed for frame-based access technologies.				
14. SUBJECT TERMS Digital video, video retrieval, video indexing, frame-based access, semantic content access		15. NUMBER OF PAGES 30		
		16. PRICE CODE		
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

NSN 7540-01-280-5500

Computer Generated

STANDARD FORM 298 (Rev 2-89)
Prescribed by ANSI Std Z39-18
298-102

DTIC QUALITY INSPECTED 4

Sponsored by
Defense Advanced Research Projects Agency
Information Systems Office

Final Technical Report
on
Video Retrieval Based on Language and Image Analysis

May 28, 1999

ARPA Order No. E994/B
Program Code No.: 7S10
Issued by DARPA/CMO Under Contract No. MDA972-97-C-0014

Name of Contractor: ADVANCE Inc.
Business Address: 15210 Enterprise Court,
Suite 300
Chantilly, VA 20151
Effective Date of Contract: 4/30/1997
Contract Expiration Date: 5/28/1999

Principal Investigator: Yiqing Liang, Ph.D.
Phone Number: 703-968-2900 ext. 729
Short Title of Work: Video Retrieval Based on
Language and Image Analysis
Reporting Period: Final Technical Report

TABLE OF CONTENTS

Abstract.....	1
1. Introduction	1
2. Technical Problem.....	1
3. General Methodology.....	2
4. Technical Objectives - Toward Semantics-based Digital Video Indexing and Retrieval.....	4
5. Technical Results.....	5
5.1 A Matured Frame-Based Digital Video Database System	5
5.1.1 Shot Boundary Detection and Keyframe Extraction.....	5
5.1.2 Algorithms Improvement for Predator Video	7
5.1.3 MPEG Applications and User Interfaces	8
5.1.4 Fusion of Multiple Modalities.....	9
5.1.5 System Architecture for Frame-based Digital Video Indexing and Retrieval.....	11
5.1.5.1 Browsing.....	11
5.1.5.2 Search and Retrieval	12
5.1.5.3 System Architecture Design	14
5.2 Toward Object-based Digital Video Indexing and Retrieval	15
5.2.1 Framework for General Object Indexing and Retrieval	15
5.2.1.1 Feature Representation.....	16
5.2.1.2 Texture Analysis	18
5.2.1.3 Decision Tree (DTs)	18
5.2.2 Special Object Indexing and Retrieval.....	19
5.2.2.1 Human Face Identification and Recognition.....	20
5.2.2.2 Regions of Military Interest.....	21
5.2.3 Object Tracking	21
5.2.3.1 Object Tracking Based on Generic Algorithms	22
5.2.3.2 Object Tracking on DC and AC Parameters from MPEG Sequence	24
5.2.3.3 Object Tracking for Face Detection.....	26
5.2.4 Object Recognition	28
6. Important Findings and Conclusions.....	29
7. Significant Developments	29
8. Implications for Future Research	30

TABLE OF FIGURES

Figure 1. Frames from a Predator video and their accumulated displacements.....	7
Figure 2. User Interface for Shot Boundary Detection for MPEG-II Video Stream.....	8
Figure 3. Edge Counts	9
Figure 4. Shot Boundary Detection Algorithm Adapted for Predator Video.....	9
Figure 5. Metadata in Predator Video.....	11
Figure 6. Table of Contents	12
Figure 7. Results of Searching Catalog Information.....	12
Figure 8. Catalog Information	12
Figure 9. Storyboard	12
Figure 10. User Interface for Similarity-base Search and Text-based Search	13
Figure 11. System Architecture for Frame-based Video Indexing and Retrieval	14
Figure 12. System Architecture of Object-based Representation in Video	16
Figure 13. Video Frame Segmented at Sub-object Level	19
Figure 14. Methodology for Face Recognition from Digital Video Database	20
Figure 15. Algorithm for Detecting Regions of Interest	21
Figure 16. AMMA System Architecture.....	23
Figure 17. Example of a Video Sequence.....	23
Figure 18. Local Flow Vector.....	24
Figure 19. After Cancellation	24
Figure 20. Target Region Found.....	24
Figure 21. The Final Result	24
Figure 22. Original Frames	26
Figure 23. Extracted DC Frames	26
Figure 24. Extracted DC+2AC Frames.....	26
Figure 25. Face Detection System Architecture	27
Figure 26. Sample Frames of two Video Sequences.....	27
Figure 27. Errormaps Corresponding to Video Sequences in Figure 26.....	27
Figure 28. Rough Face Region of Keyframes Found by Motion Detection	28
Figure 29. Final Results (boxed) of Face Detection using Color Model Built by DT	28

Digital Video Indexing and Retrieval

Abstract

This report summarizes the technical work ADVANCE Inc. performed under DARPA's contract no. MDA972-97-C-0014. This project has been conducted during an approximate two and one half year period in Phase I and Phase II implementation. The work performed has been focusing on the research and development of technologies for the indexing and retrieval of digital video content. The methodologies are built upon the theories of frame-based access and object-access, which allows users to index and retrieve video contents accordingly. Research is also performed on access at levels of video scene and video programs. For these purposes, algorithms for video shot boundary detection and keyframe extraction were developed. Closed captioning is extracted and associated with video shots so that keyword searching can be performed to retrieve video shots. Different approaches are used to segment video objects, including motion-based segmentation, and the combination of motion information and image cues for segmentation. Special objects such as human face and military target of interest were investigated as indexing points. Techniques for object identification and tracking were also studied for indexing and retrieving video contents. A comprehensive prototype has been built and software has been developed for frame-based access technologies.

1. INTRODUCTION

The World Wide Web has emerged as a huge and continuously growing distributed digital library. Intelligent and content-based information retrieval techniques are in pressing demand, especially intelligent techniques for content-based retrieval of non-textual information, particularly video information. It is this pressing demand that led to our Small Business Innovation Research contract with DAPAR's Image Understanding program, for both Phase I and Phase II.

Our Phase I contract started April 30, 1996 and was completed as scheduled on October 30, 1996. During this phase, we developed some rudimentary algorithms to extract keyframes, invented approaches to grab close captioning information from video and align it along time line with video, integrate keyframes and closed captioning information with IBM QBIC system, and built a prototype of keyframe based digital video database. Our Phase II project started on April 30, 1997 and was completed on May 28, 1999. During the Phase II project, we focused on developing new technologies that are aimed at moving our technologies from frame-based indexing and retrieval toward object-based indexing and retrieval. We also greatly enhanced our technologies for video shot boundary detection and keyframes, adapted these algorithms to military applications such as UAV Predator video, and incorporated capabilities to grab metadata information that may come from Predator video. We improved the user interface and developed our algorithms into software that can support both MPEG-I videos and MPEG-II videos. In addition, we made efforts to market our technologies to potential customers including General Dynamic Information Systems (GDIS), National Imagery and Mapping Agency (NIMA), Video Working Group (VWG), Marshall Associates Inc., Lee Technologies, etc. We also published a number of papers at national conferences (see attached publication list).

This report gives a detailed account of our technical efforts to build technologies for digital video indexing and retrieval. Section 2 presents our understanding of the importance and motivation for the development of digital video indexing and retrieval technologies. Section 3 reviews the state of the art and related works in the field. Section 4 describes our technical path and the rational behind the path. Section 5 gives detailed descriptions of the technologies we have developed. Section 6 will discuss our future work.

2. TECHNICAL PROBLEM

The recent battle over the broadband access for the Internet, as represented by the acquisition of MediaOne by AT&T further witnessed the importance of digital video contents. However, broadband communications lines to every household are not yet everywhere. The amount of digital TV and video streamed over Internet and intranets have been exploding due to the convergence of such technologies as higher-bandwidth network access, video compression, video streaming engines and players, and

multimedia-enabled PCs. Organizing this vast amount of video information to make it a searchable media is another important and urgent task. The technology to analyze, understand, index, retrieve, and visualize digital information from large video information repositories has enormous commercial potential. Potential users of such systems include:

- Scholars doing academic research; journalists searching for background information
- Professional TV producers editing video clips, creating new video contents
- Legal assistants reviewing previous testimony
- Law enforcement and criminal investigators searching through video records;
- Military analysts compiling databases on regions or topics of interest
- Security people using video monitoring and surveillance
- Coaches and trainers using videos integral to dance and athletics
- Meteorologists using satellite imagery
- Home video library/album indexing and accessing home collection of VHS tapes, CD-RWs, DVD.
- Home video diary records the daily activities in the house providing pictorial summary
- Advanced parental control filters out only parts of TV programs that may have too much skin content, violence, and obscene word, based on the contents.

Digital video database systems will find applications in almost every sector of the economy and society when computer technology advances. The economic benefits that these technologies will bring will definitely create new wealth for the American people and society. In addition, these advancements will have great impact on modern technologies and will push the technology envelope further, thus benefiting the human society.

3. GENERAL METHODOLOGY

The rapid advances in computer technology and accessibility to the Internet have led to a demand for easy access to video contents. Extensive research in the field of digital video indexing, storage and retrieval has been conducted both in the United States and in Europe, Japan and South East Asia. Progress has been made in processing digital video and capabilities of digital video databases and digital video database management systems are being extended. Universities have started to work in this field as have commercial firms.

Though there are numerous research activities going on around the world, there are only a few of these systems being in production or trial production. One of them is our FETCH system. FETCH system automatically detects shot boundaries and extracts keyframes based on video contents. It provides both browsing and searching capabilities for users to non-linearly access video contents. Browsing allows users to get to desired keyframes in a structured way by presenting video program lists, catalog information, video storyboard, or Scene Transition Graph (STG), and text search capabilities of the catalog information. Search allows users to randomly search the entire video archive to find desired keyframes using image similarity search and text search based on closed captioning. The FETCH demo system has over 40 video programs totaling over 14 hours' contents. The database contains over 800 keyframes indexed by both image features and closed captioning. It is to be deployed at Marshall Associates Inc. and has attracted interest from various organizations including Lee Technologies. Virage's VideoLogger uses image analysis techniques to slice the video into segments based on changes in the visual content, such as scene cuts, camera pans and zooms. It extracts distinct keyframes to represent each segment, generating a digital storyboard that communicates the visual subject matter in a highly efficient way. In addition to keyframe images, the VideoLogger extracts any text associated with the video signal, such as closed captions or teletext. Users can also mark and annotate clips "on the fly" or by selecting in and out points from the video storyboard. Virage has scored the best marketing achievements so far. Compaq and CNN are using their technologies. ISLIP's MediaSite Logger, MediaSite Builder, and MediaSite Finder provide the capabilities of video cataloging, automatic indexing and archiving, search, and retrieval. Their main feature is the integration of a smart speech recognition system, which enables fast video skimming, indexing, and retrieval based on speeches. ISLIP's product has the endorsement from Ford and Boeing. Excalibur and a company in Australia have software that can perform similar video analysis functions, but short of composing a complete system.

These current technologies share one thing in common, i.e., video frame is used as their primitive data unit. All the operations, manipulations, and analysis are based on these primitive data units. We classify these technical approaches as frame-based approaches.

The frame-based indexing and retrieval of digital video have found their practical applications, however, they are still far from adequate to meet users' requirements. The state-of-the-art approaches to querying, indexing, and retrieving video contents often result in cumbersome search and inaccurate retrieval. Users generally want to query the system at the semantic level rather than use such features as color histogram to describe a concept. Semantic queries such as search for particular shapes, distinct semantic objects ("man", "car," etc.), or higher level semantic structures ("anchor person in news program", "football players," etc.) are still beyond our reach. Imagine users looking at their home video album (not photo album any more!) and asking, "find all the video clips with my Dad and Mom digging a hole in the yard in front of our old house!" Current technologies cannot handle this question.

This incapability of current video indexing and retrieval technologies has its root in the approaches adopted. Frame-based technologies for video information indexing focus on shot boundary detection and key frame extraction. In these systems, the basic indexing unit is "shot" or "story unit", and visual representation of shots and/or story units is provided by a set of static "keyframes" which may be represented as scene transition graph (STG) or a "video storyboard/video poster". Data fusion is implemented only for such multiple modalities as image and text information from speech or closed captioning. These technologies are developed based on global information such as color/color histogram, edge change ratio, contrasts change, standard deviation of pixel intensities, etc. While all these efforts in frame-based indexing and retrieval have made contributions to the solution of video indexing and retrieval problem, there are several significant deficiencies with these approaches. First, global information is low level information and precludes the possibility of high-level abstraction. The result is thus lack of semantics. Second, they do not provide access to individual video objects in the video stream since frame is the smallest primitive. Third, some prominent type of information such as motion that differentiates video/motion imagery from still images is not well explored for this purpose, though motion information such as optical flow is utilized to help shot boundary detection. These deficiencies are the targets of our research.

Recently, research efforts have been directed at more advanced technologies, including semantically segmenting video objects, identifying high level semantic and logical structures of video such as Table of Contents (TOC), and presentation of video such as video summaries.

Research in semantic object segmentation has been going on for years. The entire area of automatic target recognition aims at recognizing objects and associates semantic meanings to them. Significant progress has been achieved. The applications of statistical robust maximum likelihood estimation of mixture models to optical flow at Sarnoff Corporation, Xerox, and University of Toronto have scored remarkable success in object segmentation. Morphological moving object segmentation and tracking has been invented at Swiss Federal Institute of Technology. However, all these are piecewise work. The only system level work we have seen is the work at Columbia University where morphological moving object segmentation was used first to segment objects and then semantic visual templates are used to associate semantic meanings to visual features. However, this effort is restricted to a lab environment and works only on a few simple cases.

High level semantics and logical structures reflect the original intention of the video creator, which is lost when video data is recorded on the linear media. The tasks toward the recovery of these structures include identification of the semantic structure embedded across multiple media, and discovery of the relationship among the structures across time and space so that higher level of categorization can be derived to further facilitate automated generation of a concise index table. Utilizing image cues from the video to achieve this purpose has been attempted. Scene Transition Graph (STG) invented at Princeton University uses time-constraint clustering of keyframes extracted to organize video around story units, or scenes. High level semantics can also be recovered by relying on text information. Using fixed textual phrases such as "still to come on the news ..." to exploit story boundary is such an example. Efforts are being made to use

an integrated approach to achieve this purpose where cues from different modalities and media are utilized whenever it is appropriate, including the predefined content hierarchy such as news broadcast video.

4. TECHNICAL OBJECTIVES -- Toward Semantics-Based Digital Video Indexing And Retrieval

The description of the state-of-the-art indicates the technical directions that the technologies for digital video indexing and retrieval should take. The unique characteristics that differentiate digital video processing from other information processing define the methodologies we can adopt. These unique characteristics include the large amount of information, the uninterruptible transmission of information, the linear sequential nature of traditional analog media in the form of film or analog video, and the multiple modalities of information present in video. All these characteristics present major difficulties to developing effective tools for search, browsing, navigation, annotation, etc. and bring with it such grand challenging tasks as content tools and management capabilities, in addition to such tasks as common interfaces and conversion equipment, and compression techniques. On the other hand, large amount of information, the varieties of information, and such information as motion that is absent in still image or text information, also provide us with more capabilities to better solve the problems. The key is to adopt appropriate technologies.

We define video data structure as hierarchical layers of information including:

- Video program/clip
- Video Scene: a collection of one or more adjoining shots that focus on an object or objects of interest
- Video Shot: an unbroken sequence of frames from one camera operation; boundary changes: cut, fade, dissolve, and wipe
- Video Objects that satisfy spatio-temporal constraints and Video Classes that are the abstraction of Video Objects and bear semantics.
- Video Regions of Interest that have coherent and consistent features over the regions
- Video Image Features
- Video Image Pixels

The last three levels of information are the primitives that constitute the basis of other levels. They usually do not have explicit semantic indications and are not the direct target for human attention. The first four layers are what people are usually concerned about. Extensive research and commercial work has gone into the use of first three (3) layers and significant progress has been achieved. Apparently, each of these layers constitutes a step towards human access to the contents of video.

Another aspect in video access is the multiple modalities of information embedded in video. These multiple modalities include image, audio track (speech and special effects), closed captioning, caption, metadata, motion, and spatio-temporal relation. All these modalities of information are closely related to the semantic contents in video and should contribute to the indexing and retrieval of video contents.

Multiple Modality	Multiple Levels
Image	Video Programs/Clips
Audio Track <ul style="list-style-type: none"> • Speech • Special Effects (music, laughing, gunfire, explosion) 	Video Scene a collection of one or more adjoining shots that focus on an object or objects of interest
Closed Captioning	
Caption	Video Shots
Metadata	an unbroken sequence of frames from one camera operation
motion	
Temporal Information	Video Objects

We believe that making synergetic use of these multiple modalities is the correct approach to developing technologies for the access to video contents.

5. TECHNICAL RESULTS

Following the technical path discussed in Section 4, we focused on building a prototype of a digital video database and developing preliminary algorithms for accessing video contents at frames level in our Phase I effort. During Phase II, we built a comprehensive trial database system based on video frames and developed matured technologies around it. We also opened the door toward object-based access and performed advanced research in terms of detecting video objects and identifying them as indexing and retrieval points.

5.1 A Matured Frame-Based Digital Video Database System

At ADVANCE, we have built a comprehensive and practical pilot system that allows frame-based access to digital video contents. This system has included many functions including:

- Video keyframe extraction using motion and intensity information;
- Video content presentation using keyframe storyboard or Scene Transition Graph;
- Video keyframe searching based on image similarity and
- Video keyframe searching based using full text retrieval where text is generated from closed captioning.

5.1.1 Shot Boundary Detection and Keyframe extractions

One popular approach to digital video indexing and retrieval is to divide a long video stream into segments called shots since shots are the basic components of a video program. This segmentation is usually implemented through shot boundary detection. Once shot boundary is detected, keyframes are selected from the detected shots to represent the contents of that shot. Various methods of automatic shot boundary detection and keyframe selection have been investigated. These methods include:

- Absolute frame difference – the measure of the difference between two consecutive frames of the sum of the color/luminance intensities of all pixels in the frame.
- Color/luminance histogram-based: either for the entire frame or using a set of histograms to capture object structure of the frame.
- Simple histogram difference
- Weighted histogram difference
- Histogram difference after equalization
- Intersection of histograms
- Shared histogram difference
- Based on moment invariants, which are invariant to scale, rotation, and transition.
- Based on the range of pixel-value changes, i.e. the difference between the values of a pixel in two consecutive frames as the combination of noise generated by camera and digitizer, motion of objects and camera, and changes caused by cuts and other transitions.
- Based on edge detection, which calculates the difference of the percentage of edge pixels in frame f which are more than a fixed distance r away from the closest edge pixel in a frame and the corresponding percentage in the consecutive frame.
- Based on encoded information, which can be obtained from such compression information as Discrete Cosine Transform (DCT) coefficients.

Keyframes are often selected as the first frame or 1st frame of the shots. Though these algorithms can perform shot boundary detection well to a certain extent, they all need further improvement. A recent study by Intel calls strongly for algorithms that employ motion information to achieve these desired enhancements. This call is based on the observations that all current algorithms for shot boundary

detection are negatively influenced by global and local motion in video, and therefore identification of shot boundary relying on motion is encouraged.

We are very fortunate that we have already developed an algorithm that makes good use of global motion information, together with intensity information.

First, we use a metric of motion to judge whether a shot has occurred. In addition, we apply hysteresis to suppress cuts, which appear too closely spaced. The result is a stream of transitions which has a high true positive rate but an unacceptably large false positive rate. We then apply a stuttering elimination phase, driven by a luminance histogram metric, to reduce the number of false positive transitions. Based on the resulting transitions, we can select keyframes to represent the shots. The thresholds used in each detection step can be specified at run-time by the user. We have found that some video programs require different thresholds, but that most programs can use default threshold values we determined by experiments.

We use a new metric, motion pseudo-variance, as a metric for selecting an initial set of candidate cuts. This analysis is based on the optical flow of the frame, which provides more accurate estimates of motion than using MPEG motion vectors. We compute $M(t)$, the total amount of motion in frame t by first using Horn and Schunck's algorithm⁴ to compute an optical flow vector for each pixel. If i is the x component of optical flow at pixel i, j in frame t , we define the motion pseudo-variance of a frame as:

$$MV(t) = \frac{\sum_i \sum_j |o_x(i, j, t) - o_x(i, j, t-1)| + \|o_y(i, j, t) - o_y(i, j, t-1)\|}{npixels} \quad (1)$$

The metric takes advantage of the fact that motion is generally coordinated from frame-to-frame within a cut, but optical flow provides unpredictable changes at each pixel during a cut. The higher the value of motion pseudo-variance, the more likely that the computed changes in pixel-by-pixel motion were caused by an abrupt transition. The value of $MV(t)$ is compared to a (user-definable) threshold to determine whether a candidate cut has been detected.

We add hysteresis to the transition detection system to eliminate some false positives early in analysis. We suppress transitions for a given number of frames immediately after a transition is detected. After the hysteresis period is over, detection reverts to normal operation. The current hysteresis value is 5 frames (at 30 FPS) for cuts. Hysteresis is essential when a motion-derived metric is used to detect transitions, since the numerical computation of motion requires a few frames to settle down after an abrupt transition. Even highly-edited programs, such as commercials, are unlikely to contain shots of such short length. In addition to hysteresis for cuts, we also trigger hysteresis to suppress camera flashes. Flashing lights from cameras are common in news coverage. They cause a single-frame peak in luminance. We detect these single peaks and use a 2-frame hysteresis interval to suppress false transitions caused by flashes.

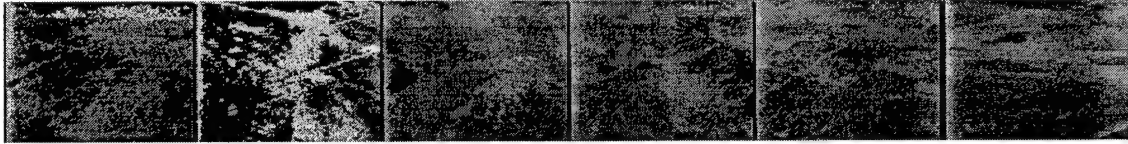
While the metric can be tuned to find most true cuts, they also generate some false positives. We use the luminance histogram difference as a test for rejecting false transitions. Using different metrics for detecting transitions and for suppressing false positives increases the robustness of our technique. A luminance histogram consists of N bins; each bin represents a range of pixel luminance, with all bins having the same luminance range. The luminance histogram function $L(i)$ is equal to the number of pixels in the frame which fall into the i^{th} luminance range. We compute the difference between two luminance histograms L_1 and L_2 as follows:

$$\frac{\sum_{1 \leq i \leq N} |L_1(i) - L_2(i)|}{npixels} \quad (2)$$

5.1.2 Algorithms Improvement for Predator Video

Predator video is quite different from news video or entertainment video where artificial shots are abundant to express authors' ideas. Predator video camera often takes video by scanning a vast background and then focuses on specific target areas. In addition to the more noises brought in by unsteady and high speed flight motion of the UAV, the transmissions of video, and natural and often un-ideal shooting environment, Predator video suffers from the lack of internal structures –artificial cuts, as compared with entertainment video or news video. There is almost no artificial cut, but many graduate transitions in the background. This creates new challenging tasks for detecting shot boundaries. Normal shot boundary detection and keyframe extraction algorithms will not apply well to Predator video and will miss a lot of scenes that might be of interest. This is an important and practical issue in real application video but has not drawn enough attention from the research community. We have been working to adapt the current algorithms for shot boundary detection for Predator video applications with two directions: taking into consideration displacement and edge counts.

We designed the improvements on the algorithms by using cumulated displacements and edge counting. We view that while the camera scans the background, the changes in the image accumulate. The extent of change accumulations can be indicated by their accumulated displacements from previous frames. When background's accumulated changes are large enough, it indicates another important indexing point that can be represented by a frame extracted at that point. These displacement measurements reflect the corresponding geographic changes in the background and thus can be used to divide this segment and to extract keyframes to represent the contents. However, accumulated displacements may not bear any significant meanings from the scenes. Adding another measurements of edge counting will assign more semantics because edge counting indicates potential objects in the scenes.



	frame 1267	frame 1349	frame 1403	frame 1457	frame 1512	frame 1581
displacement (u)		4.972462	5.455323	5.618622	4.988975	4.730186
$a0$		9.036143	10.029892	9.995114	8.508655	6.916816
$a0/u$		1.817237216	1.83855148	1.778609664	1.705491609	1.462271462
frame no. in between		82	54	54	55	69

Figure 1. Frames from a Predator video and their accumulated displacements

The displacement measurements are obtained through robust statistics for motion analysis or optimization solution for multi-scale image processing problems (see details in next section). Considering the motion caused by camera movement is the major motion in the scene, we use affine model to model the motion as defined in the following equation

$$u(x, y, a) = \begin{bmatrix} u(x, y) \\ v(x, y) \end{bmatrix} = \begin{bmatrix} a_0 + a_1x + a_2y \\ a_3 + a_4x + a_5y \end{bmatrix} \quad (3)$$

If we take the mean value of translational motion u , we have:

$$\overline{u(x, y)} = \overline{a_0 + a_1x + a_2y} \quad (4)$$

Considering x and y are averaged over a fix value range (the horizontal size and vertical size of a frame) and their average should approximate a constant value, so should the average value of u . Thus measuring a fix value of u should indicate a complete new scene.

5.1.3 MPEG Applications and User Interface

Implementation of these algorithms has been carried out for both MPEG I video and MPEG II video. MPEG I video has been widely used in the last 5 years or so as a digital video standard. MPEG II standard is getting more momentum recently since future applications in HDTV, Digital TV, and DVD have all adopted MPEG II. More and more US military units are tilting toward MPEG II. An important military user of digital video – National Imagery and Mapping Agency (NIMA) is a very strong MPEG II advocate. Thus, we have made efforts to make sure that our implementation of these algorithms will support both of these standards. As we move to the future, we see that whatever digital video standard that is popular on Internet should be supported.

Displacement

Detecting scene boundary and extracting keyframes is the basic task in video indexing and retrieval. Creating a nice user interface to make it more meaningful for the user is of great importance. We have improved the user interface, which allows the programs to be run in either batch mode or an interactive mode. In batch mode, the program can be executed faster, however, nothing is displayed while all these activities are carried out and user can only see it after all programs are finished executing. In interactive mode, user is able to see a window playing back the video while another window displays all the keyframes extracted in a montage mode. The program will be executed at slower speed, however, the user will be able to see if the keyframes are correctly extracted, if there is any important keyframe missing or if there is any redundant keyframes. This will boost user's confidence in the system and user's acceptance of the system, and also enable user to correct system's errors if needed. In both modes, the system provides user options such as sampling rate to fine-tune system performance. A sample user interface is shown in Figure 2.

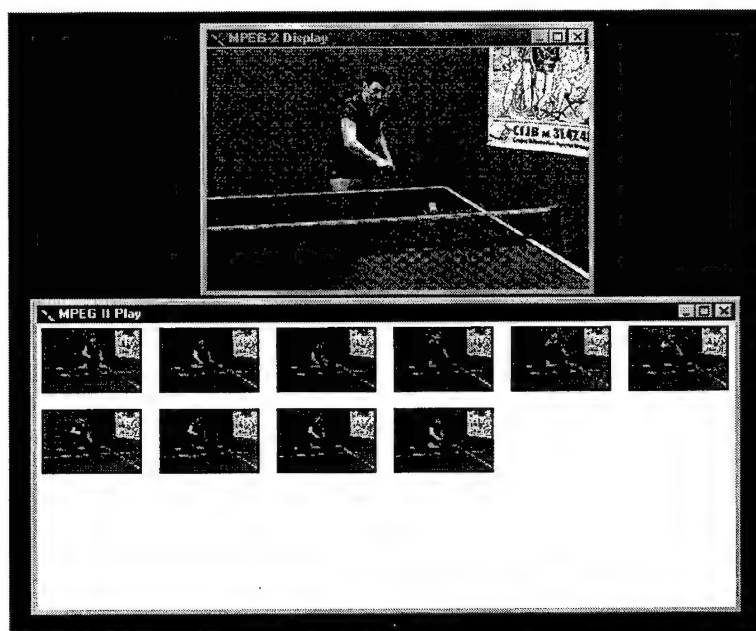


Figure 2. User Interface for Shot Boundary Detection for MPEG II Video Stream

Edge Counts

We also have incorporated edge detection algorithms that can help count edges in each frame. Some sample edge counts are shown in Figure 3. The shape of edge counts changes in the figure does indicate some rules that we might be able to explore for our shot boundary detection and keyframe extraction purposes. For example, there are local max or minimum values of edge counts between the frames representing complete new scenes.

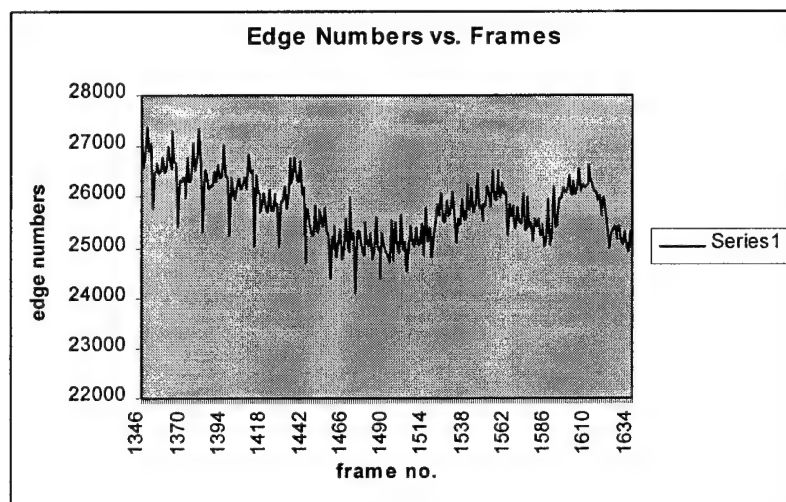


Figure 3. Edge counts

We combine displacements and edge counts, and fine-tune these variables for the purpose of extracting good keyframes set. The new algorithms adapted for Predator video can now be represented by Figure 4 to take care of displacements and edge counting.

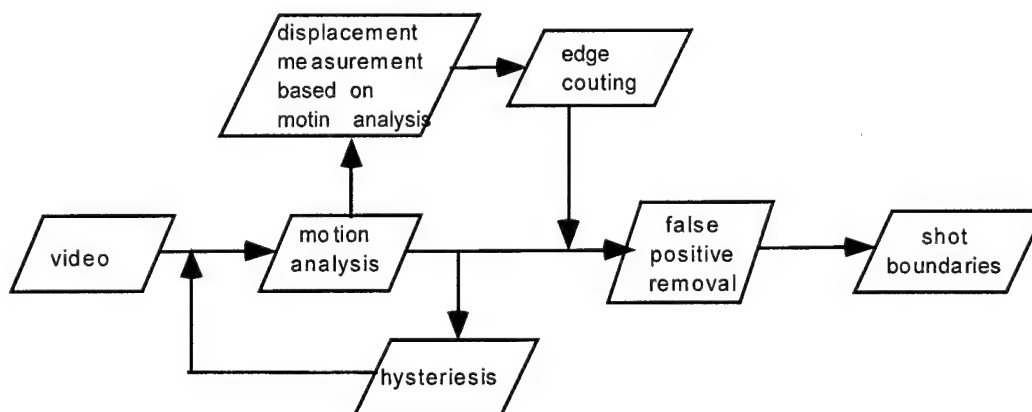


Figure 4. Shot Boundary Detection Algorithms Adapted for Predator Video

5.1.4 Fusion of Multiple Modalities

We have explored approaches to fusing different modalities of information from video to achieve better results of indexing video shots. The rational behind it is that the better we make computers understand video contents, the better videos can be automatically indexed. Toward this direction, we have made efforts to use text information embedded in closed captioning channel from video since text information is

not only part of video contents, but also helps to interpret video image contents. There are two types of text information coming from this channel: text transcripts of speeches/dialog/naration that accompany video and metadata.

Among the multiple tracks of video information, many can be presented in text format. These include audio track for speech/dialog/naration, closed captioning information, caption, and sound track information such as music and songs, in addition to the catalog information as the ones in libraries. We studied the feasibility to extract text transcripts of speech/dialog/narration from audio track. We did research on speech recognition technologies that can be used for this purpose. However, for the purpose of ease of use, we opted to use closed captioning channel to extract text transcripts.

Many commercial packages for speech recognition were reviewed including ones in the research effort. Special attention was paid to packages capable of reading from MPEG files in order to recognize speeches in view of our plan to use MPEG as the digital video standard and the fact that MPEG files for video and audio can be generated at digitization time. None of the reviewed packages can meet the requirements defined above.

Catalog Information

Available cataloging information was investigated for additional data fusion. Cataloging information for video clips is generally available at every video archiving place, although predominately entered manually. This information usually briefly describes what each video programs is about and includes title, author, the date of production, provenance, description, and keywords of each vide program. It is very useful when users are looking for video programs through text searching. However, it might take at least as long as the video program is for librarians to catalog this video program after they review it and record this descriptive information in a database.

Closed Caption

Extracting text information from closed captions in the video is another approach to increase indexing and retrieval keys for video frames. We studied the off-the-shelf software/hardware products capable of extracting closed caption information from video, and selected to use SoftTouch's equipment for this project. SoftTouch's HUBCAP is an economical stand alone caption and XDS decoder, caption and XDS data recovery, and caption character generator device. The test indicates closed caption text can be extracted and time stamps grabbed.

These time stamps can assist in building a relationship between video frames and closed caption text. Complications are anticipated in such a relationship. A closed caption statement may span many frames, even more than one shot period. Conversely, one video shot that is represented by one key frame may correspond to multiple closed caption text sentences. The topic upon which the research will be conducted is how to build such a relationship. Currently, we employ a straightforward algorithm, i.e., matching the time stamped sentences with the keyframes that have corresponding time stamps.

Metadata

Metadata defines the data embedded in video and is being considered by MPEG 7 committee for possible inclusion. NIMA is a also strong advocate of this data field. In addition to definition of data fields in video, it also contains many important fields about geo-physical information and video camera information. These fields are often needed in 3-D modeling from video, geo-registration, video mosaicing, and database population. Inclusion of metadata as indexing points is thus of vital importance.

We have done two (2) things to promote the incorporation of metadata for video indexing and retrieval. First, we succeeded to extract metadata from video tape while it is digitized so that we can have access those data. Second, we have designed algorithms and implemented the algorithms in Perl language to locate these data elements from the closed-captioning data stream and also associate it with keframes. This enables us to provide the capabilities to let user to

- search specific video frames or video shots using these specific metadata terms, or

- provide these metadata when user searches and locates video frames or video shots through video database search

Our experiments are performed on metadata that comes with the Predator EDS Demo tape, Ver Beta-1, 29 Sep 97, from Pete Wiedman. We compared it with Video Metadata Dictionary Version 0.1.3, November 17, 1997 as drafted by Video Metadata Group, a subgroup of Video Working Group. The analysis and comparison shows that the metadata embedded in the tape is only a subset of the metadata defined in Metadata Dictionary as shown below:

15:43:38.35	+34°38'56"	EO Zoom	6,120MSL
	(6.13 vehicle latitude)	(6.8 sensor ID)	(6.12 vehicle altitude)
	-117°37'05"2	+1.76°	17.18°
	(6.14 vehicle longitude)	(sensor depression angle)	(6.3 sensor FoV angle)
	+00°00'00"	16:28:59	0ft
	(6.4 image center latitude)	(time/data(alternating))	(6.20 slant range)
	+000°00'00"	269.36°	0ft
	(6.5 image center longitude)	(payload azimuth)	(gnd distance AtImageBase)



Figure 5. Metadata in Predator Video

The numbers in parenthesis are section numbers as defined in Video Metadata Dictionary. The figure below shows how the metadata is displayed with video playing back.

Though we have not incorporated this capability into our digital video database system yet, the general problem of search specific data elements in metadata is considered resolved. It would be better to find what will be user's access requirements through metadata before we move further.

5.1.5 System Architecture for Frame-Based Digital Video Indexing and Retrieval

Nowadays, the most popular approach for people to access information on the Internet is through browsing and search. We followed this pattern and designed the system to provide both the functions of browsing and search for users to access video contents.

5.1.5.1 Browsing

Browsing mechanism leads users in a structured way to arrive at the video contents users are looking. It allows people to browse and navigate through our digital video library. Users can select to enter our digital video library and view the table of contents that list all video programs in the database. They can select one of the programs by clicking on the corresponding item in the table of contents. Also, users can apply text search based on the catalog information to retrieve corresponding programs with key words when the list gets too large to review.

Once they get into a particular program, they will see all cataloging information. Users have several options: playing back the entire video program; playing back only audio part; viewing the video storyboard; or viewing the STG. Upon selecting storyboard, user will see a storyboard that is composed of all the

keyframes extracted from the video. In this way, users don't need to spend the time going from the beginning to the end of a video program to find which segment they want. All they need to do is to look at the storyboard, which summarizes the contents of video. User can click on a particular keyframe that represents a shot. The user will then be able to view an enlarged frame, playing back only that video shot, and viewing the text spoken during that shot.

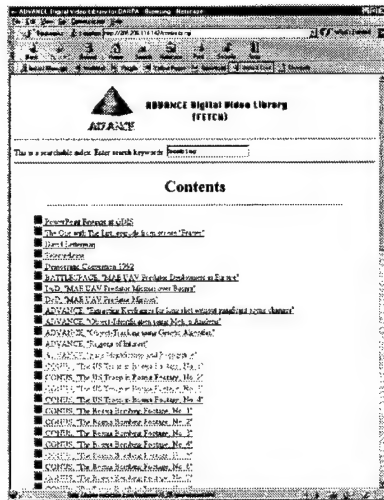


Figure 6. Table of Contents

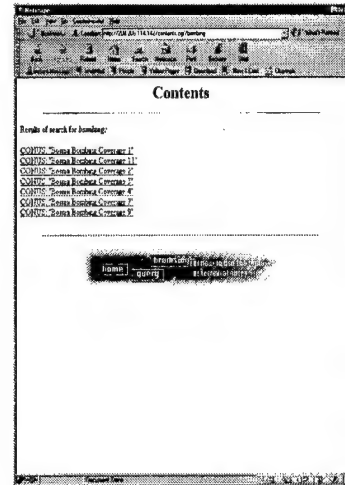


Figure 7. Results of Searching Catalog Information

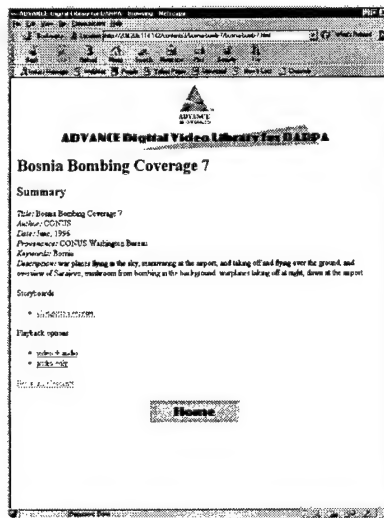


Figure 8. Catalog Information

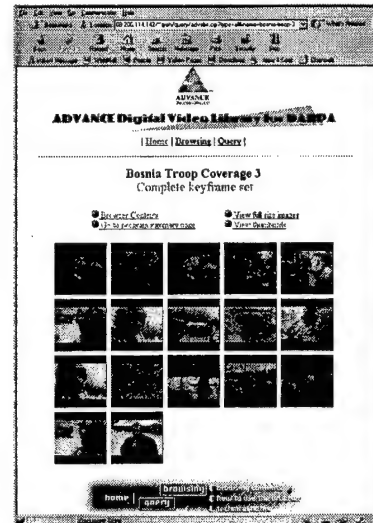


Figure 9. Storyboard

5.1.5.2 Searching and Retrieval

Video retrieval is concerned with video sub-sequences (shots and segments) and video keyframes that are related to the user's query request. We have designed a database indexing and retrieval mechanism based on the keyframes. Since keyframes capture the most important contents in video programs, retrieving video shots could be conducted by retrieving key frames representing that shot. If we organize all the video key frames from all video programs in one single database, a retrieval of certain keyframes should allow us to access all video contents in a non-linear manner to the level of video shots. Consequently the problem of retrieving certain video shots is reduced to retrieving certain video key frames. Each video keyframe can be a still image, possibly a JPG file or a DC image. Thereby retrieving a desired still image is the approach capable of solving the problem of accessing video information.

Video retrieval systems can be implemented as text-based and/or content-based. We implemented text search approach to accessing video contents as discussed in next section. For content-based access, we implemented interactive and similarity-based approach as well as subject-based approach.

Searching based on image similarity

Currently, we employ IBM's Query By Image Content (QBIC) (Flicker 1995) as our image search engine based on similarity measurement. QBIC is a set of software routines providing functions to query collections of images by content, thereby permitting an image collection to be queried for images that have predominantly red colors or striped textures where the color and texture information has been automatically computed. Originally, QBIC was designed for still image search. It is our belief that QBIC's still image search engine can be applied to the video key frame search. Although QBIC is restricted by its search capabilities in terms of color, color histogram, and texture, it should be capable of providing a starting base for the video database search. Every effort will be made to add our own features to QBIC, thereby gradually adding more functionality to the search engine.

QBIC search engine is built upon the theory of similarity measurements. There are many different kinds of similarity measurement algorithms: Vector Space model; Metric Space model; and More General models or the algorithms can be classified into: Metric measure; Set-Theoretic based measures; and Signal Detection Theory based measures. Each group is further subdivided into; a) measures based on crisp logic, and b) measures based fuzzy logic. When applied to keyframes in video database, QBIC grabs the keyframes image features as indexing values in terms of color, color histogram, texture, etc. and tags keyframes with these values. The system displays 10 keyframes randomly selected from the database for user to select one as most closely similar to what he/she wants. The system will search based on that one and return another 10 keyframes with similar ranking. The process will iterate until desired ones are found. Following are two windows showing the randomly selected 10 keyframes and the results of one search.

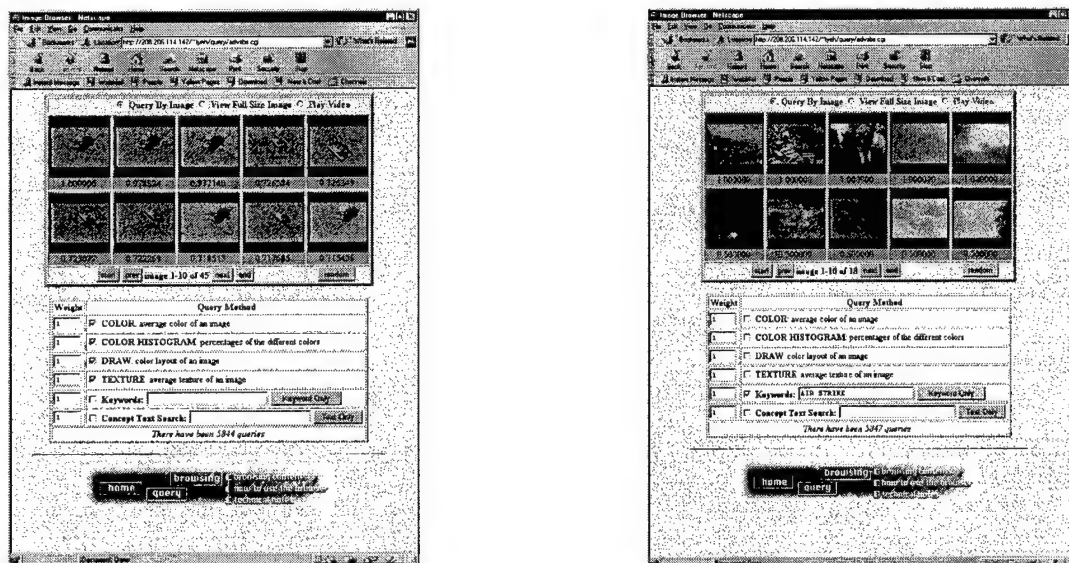


Figure 10. User Interface for Similarity-based Search and Text-based Search

Text Search Engine

The key technology to make the best use of all text information extracted from video is a text search engine. Two text search engines are used in our system to carry out matching search for image: one exact key word match from IBM QBIC, and one based on Latent Semantics Index (LSI). The text search used in IBM QBIC employs matching algorithms. On the other hand, LSI employs similarity search by concept space and concept mapping to expand the algorithms' capabilities to resolve the famous synonymy and polysemy problem in information retrieval.

Before we could use these text search engines to search for video shots, we need first to establish a corresponding relationship between text information extracted from audio, closed captions, captions, and cataloging information, with video shots, especially keyframes representing those shots. Once this relationship is established, text associated with a shot/keyframe can be segmented and treated as a document. The problem of retrieving a shot meeting text retrieval criteria becomes a problem of retrieving the document associated with the shot/keyframe. Thus, video shot/keyframe retrieval through text becomes an information retrieval problem.

Latent Semantic Indexing (LSI) (Deerwester et al. 1990) is a vector space information retrieval method demonstrating improved performance over traditional vector space techniques utilized in Salton's SMART system. LSI uses singular-value decomposition allowing re-arrangement of space to reflect major associative patterns in the data, thereby ignoring smaller, less important influences. Position in space then serves as a new kind of semantic indexing. Retrieval proceeds through use of the query terms to identify a point in space, and return documents in its neighborhood. LSI is suitable for document retrieval with sparse indexing information as is the case in video where text information is sparse with respect to image information. We made efforts to integrate LSI with IBM's Query By Image Contents (QBIC) still image database to make it a complete functioning system.

5.1.5.3 System Architecture Design

System architecture design is a key to the success of this project since the system will incorporate a number of independently developed research systems as its components. These system components include: algorithm to segment video, algorithms to extract keyframes, algorithms for rough classification and object recognition for subject-based search, algorithms to generate STG, algorithms to extract closed captioning information, algorithm to match text with keyframes, image search engine, and text search engine.

The system architecture design is shown in the following figure.

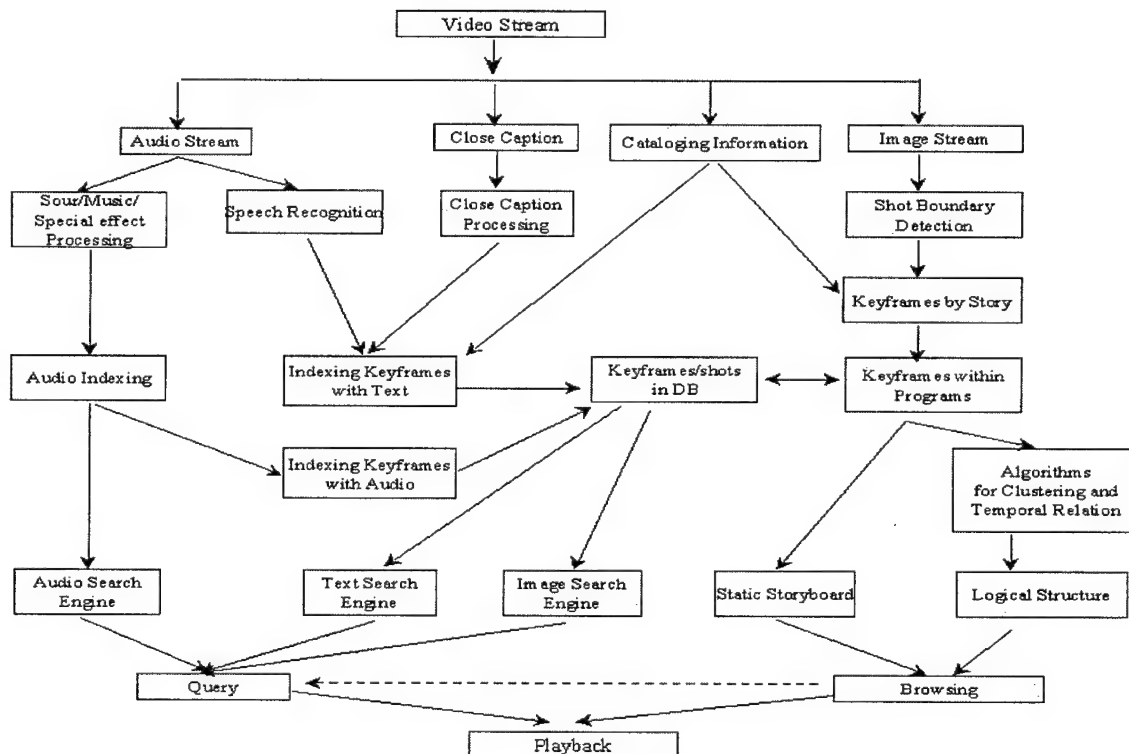


Figure 11. System Architecture for Frame-based Video Indexing and Retrieval

In addition, we take care of other factors that have important impacts on system performance.

Networking (Internet / Intranet). We use WWW navigator as the user interface and network computing as part of our fundamental system architecture. It works both on Internet and on Intranet.

Off-line Processes. The functions to create database, including extracting key frames from video streams, tagging image database, extracting text and mapping them to key frames, generating indices for key frames, and generating storyboards, are off-line processes. They will be performed without users' involvement as part of pre-processing.

Real-time Processes. The interactive functions, including submitting queries, searching the database, returning search results to the users, and playback of video clips, are real time and on-line operations.

Internet Navigator Plug-in for MPEG Video Shot Playback. Many commercially available MPEG player plug-ins can playback MPEG video file on the web browser. However, none can specify the start frame and stop frame for a preset playing-back segment. We have developed an MPEG plug-ins, which allow the start frame and stop frame be specified for playing back only the pre-selected segment of the video story. This plug-in has an 32-bit version which runs in Microsoft Windows NT and can be used for any 32-bit MCI-compliant MPEG decoder, and a 16-bit version which runs on Window95. This function give user a warm feeling about the system when they see the live playback of specified video segment, not the rest which he/she may not want to see.

Thumbnail and Full Size Picture; Browsing and Searching. Functions for storyboards and image search employ thumbnails for presentation. We support viewing the full size keyframes by just clicking on a thumbnail. Connections between Browsing and Searching are built to allow users to go easily from one approach to another.

MPEG Editing Tool. We made efforts to convert a shareware of MPEG editing tool into production software, but gave up due to the amount of work and its applicability since there are already many commercial tools for the same purpose.

5.2 Toward Object-Based Digital Video Indexing and Retrieval

Psychology and other sciences have proved that video objects are the basic understanding block in human perception and understanding of the world. Objects are the semantic primitives in video contents. Good indexing and retrieval of video contents will only be feasible once video contents are better understood by computers. Identification of objects and recognition of their meanings are absolutely necessary steps toward this understanding, and thus most important ingredients in indexing and retrieving video contents. Complete understanding automatically video contents is still far away from commercialization and identification and recognition of video objects still remain largely a research issue. However, we created the foundation for the future and performed some valuable work along this line. We have achieved certain positive results and developed a general framework for video object indexing and retrieval. We have also made progresses in making special video objects such as human face and military targets as indexing points. We have applied different approaches and implemented different algorithms for tracking video objects. We consider object tracking important because it will allow us better detect objects and thus better recognize them. We also made a trial effort for automatic target recognition.

5.2.1 Framework for General Object Indexing and Retrieval

The first thing we need to do for indexing video is to detect and recognize those video objects. We realize that detection and recognition of video objects are not easy tasks, we set out first to define a general framework that will serve this purpose best. Considering that there are multiple modalities of information existing in video, make good use of all these modalities should constitute a good approach.

We have defined our general framework for object-based video processing, as based on a representation of video contents by video object in frames. Salient video objects are segmented from frames. These video objects may correspond to meaningful real-world objects such as cars, vehicles, and humans, or low-level

image regions with uniform features, such as color, texture, or shape. Features of these objects and low-level regions are extracted to index them. These features could be color, texture, shape, and motions. For example, color may include single color, average color, representative color, color histograms, and color pairs. Texture may include textures histogram, Tamura texture, wavelet-domain textures, and Laws Filter-based texture. Shape may include geometric invariants, moments of different orders, polynomial approximation, spline approximation, algebraic invariants. These object and low-level regions are tracked over time, temporal features such as trajectories of centroid of each object or low level region, motion patterns, and life spans are indexed. The low-level regions may in turn be used to develop into a higher level of indexing that includes links to conceptual abstraction of objects. These object, low-level regions, and their features extracted from video constitute primitive indexing points, which may contain semantics such as indicated by those meaningful objects.

Once video representation by the video object model as described above has been achieved, we can build a search and retrieval system on these indexing values for video contents. A convenient query form is needed and a search engine that can match the query against these index values will be further developed. Obviously, the first important step described above is to segment video objects or low-level regions, and extract features from them. We have defined and experimented the following general framework for this purpose. This is a 3-step procedure. The region map representing the different object regions in video frames could be generated via the three steps of analysis - motion, color, and texture. The system architecture is shown in fig 12.

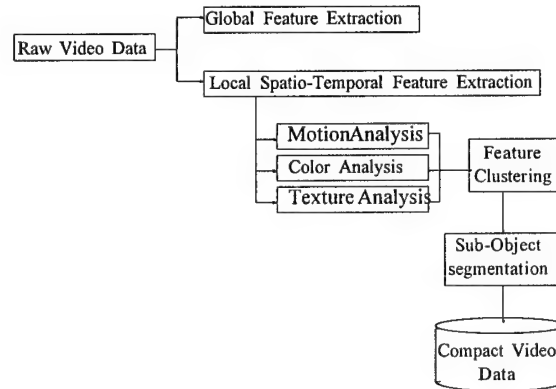


Figure 12. System Architecture of object-based representation in video

Optical flow vectors are first computed based on the motion information between two adjacent frames. The flow vectors are then clustered into 4 clusters that can partition the entire image frame into sub-regions belonging to corresponding classes. Color information consisting of normalized r-g-b space and L-a-b space as well as the texture features are also clustered and used to refine the object segmentation. The texture analysis techniques are applied on 8x8 sliced windows of each frame to produce the entropy and coarseness information. Autocorrelation function (ACF) is used here to measure the coarseness representation of two horizontal pixels - M02, two vertical pixels - M20, one diagonal pixel - M11, and two diagonal pixel - M22.

5.2.1.1 Feature Representation

Several image processing and classification techniques are used to implement both face detection and object-based representation system. These methods covering motion detection, color analysis, and texture analysis and Decision Tree (DT) are briefly introduced in following sub-sections.

Optical Flow for Motion Detection

Image motion results from the projection of an object's 3-D motion onto a 2-D image plane. The so-called optical flow (image flow) is the apparent motion of an image pattern in the image plane and it corresponds

to a velocity field. It is well known that people can easily perceive and track object's motion. The perception of visual motion involves two types of computations, those of temporal changes and spatial integration. The well-known intensity gradient model proposed by Horn and Schunck [1980] applies differential operations to both the spatial and temporal dimensions. Since natural images are not always differentiable, the intensity gradient model usually requires pre-smoothing of images. Therefore, the intensity gradient model includes a spatial smoothing filter followed by a time differentiation. Assuming velocities vary smoothly everywhere, Horn and Schunck use a global smoothness constraint to resolve the optical flow constraint equation

$$E_x u + E_y v + E_t = 0. \quad (5)$$

This equation is called the optical flow constraint equation, where the spatial and temporal derivatives (E_x , E_y), and E_t are estimated from the video sequence. As the optical flow constraint equation is over-determined it can be solved in the least squared error sense using the global smoothness assumptions mentioned above by minimizing the error function

$$Error(u, v) = \|E_x u + E_y v + E_t\|^2. \quad (6)$$

r-g-b and L-a-b Color Spaces

Face color representation can be invariant after normalizing color values from RGB space to r-g-b space [Yang and Wailbel, 1995].

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B} \quad (7)$$

The transformation from RGB to L-a-b [Sun, et. al, 1997] are:

$$L = 116 \times \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16, \quad (8)$$

$$\text{for } \frac{Y}{Y_n} > 0.008856, \quad L = 903 \times \frac{Y}{Y_n},$$

$$\text{for } \frac{Y}{Y_n} \leq 0.008856,$$

$$a = 500 \times \left(f\left(\frac{X}{X_n}\right) - f\left(\frac{y}{y_n}\right) \right) \quad (9)$$

$$b = 200 \times \left(f\left(\frac{y}{y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right) \quad (10)$$

where

$$f(t) = t^{1/3}, \quad \text{for } t > 0.008856$$

$$f(t) = 7.787 \times t + \frac{16}{11} \quad \text{for } t \leq 0.008856$$

$$X_n = 0.951, \quad Y_n = 1.000, \quad Z_n = 1.089 \quad \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.431 & 0.342 & 0.178 \\ 0.222 & 0.707 & 0.071 \\ 0.020 & 0.130 & 0.939 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

5.2.1.2 Texture Analysis

Texture is observed in the structural patterns of surfaces of objects. A textel (basic texture elements) contains several pixels, whose placement could be periodic, quasi-periodic, or random. In nature, textures are random and suited for statistical characterization. Among the statistical methods we use the Entropy and autocorrelation function (ACF) for the regular/irregular and coarse/fine measurement.

Entropy

Histogram features are based on the histogram of region of the image. Let u be a random variable representing a gray level (L gray levels in total) in a given region of the image. Define

$$p_u(x) \triangleq \text{prob}[u = x] \\ \cong \frac{\text{number of pixels with gray level } x}{\text{total number of pixels in the region}}, \quad (12) \\ \text{where } x = 0, \dots, L-1$$

The *entropy* (H) of region is defined as the average information generated by the region histogram.

$$H = E[-\log_2 P_u] = -\sum_{x=0}^{L-1} P_u(x) \log_2 P_u(x) \quad (13)$$

The Autocorrelation Function (ACF)

The spatial size of the texels in texture can be represented by the width of the spatial ACF $r(k, l) = m_2(k, l) / m_2(0, 0)$. The measurement of the texture coarseness can be proportional by the spread of ACF which are obtained via the moment-generating function

$$M(k, l) \triangleq \sum_m \sum_n (m - \mu_1)^k (n - \mu_2)^l r(m, n) \quad (14)$$

$$\text{where } \mu_1 \triangleq \sum_m \sum_n m r(m, n), \mu_2 \triangleq \sum_m \sum_n n r(m, n)$$

$r(m, n) = m_2(m, n) / m_2(0, 0)$ as we mentioned above while m_2 is the *mean square value* or *average energy* of the histogram and it is a case of the moments m_i when $i=2$. The i th moments can be expressed as:

$$m_i(k, l) = \frac{1}{N_w} \sum_{(m, n) \in W} [u(m - k, n - l)]^i \quad (15)$$

where $i=1, 2, \dots$ and N_w is the number pixels in the window W .

5.2.1.3 Decision Trees (DTs)

The basic aim of any concept-learning symbolic system supporting pattern recognition and classification is to construct rules for classifying objects given a *training set* of objects whose class labels are known. The objects belong to only one class and are described by a fixed collection of attributes, each attribute with its own set of discrete values. The classification rules can be derived using C4.5, the most commonly used algorithm for the induction of decision trees (DT) [Quinlan, 1986]. The C4.5 algorithm [Quinlan, 1993] uses the entropy as an information-theoretical discriminating measure for building the decision tree. The entropy is a measure of uncertainty ('ambiguity') and characterizes the intrinsic ability of a set of features to discriminate between classes of different objects. The entropy E for a feature set $\{f\}$ is given by

$$E(f) = \sum_{k=1}^n \sum_{l=1}^{m_k} \left[-x_{i,k}^+ \log_2 \left(\frac{x_{i,k}^+}{x_{i,k}^+ + x_{i,k}^-} \right) - x_{i,k}^- \log_2 \left(\frac{x_{i,k}^-}{x_{i,k}^+ + x_{i,k}^-} \right) \right] \quad (16)$$

where n is the number of classes and m_f is the number of distinct values that feature f can take on, while $x_{i,k}^+$ is the number of positive examples in class k for which feature f takes on its i^{th} value. Similarly $x_{i,k}^-$ is the number of negative examples in class k for which feature f takes on its i^{th} value. C4.5 determines in an iterative fashion the feature, which is the most discriminatory, and then it splits the data into two sets of classes as dichotomized by this feature. The next significant feature of each of the subsets is then used to further split them and the process is repeated recursively until each of the subsets contain only one kind of labeled ('class') data. The resulting structure is called a decision tree, where nodes stand for feature discrimination tests while their exit branches stand for those subclasses of labeled examples satisfying the test. An unknown example is classified by starting at the root of the tree, performing the sequential tests and following the corresponding branches until a leaf (terminal node) is reached indicating that some class has been decided on. After decision trees are constructed a tree pruning mechanism is invoked. Pruning is used to reduce the effect of noise in the learning data. It discards some of the unimportant sub-trees and retains those covering the largest number of examples. The tree obtained thus provides a more general description of the learned concept.

Fig. 13 shows the sub-object regions in Keyframes that are segmented successfully. The white-line indicates the regions segmented by motion analysis while the gray-line indicates the regions wrapped by color and texture analysis.

Note that those regions can be represented by their corresponding cluster centers and can be indexed for future retrieval. Since the clustering centers represent the abstract content information of scenes, the semantic description can also be added on.

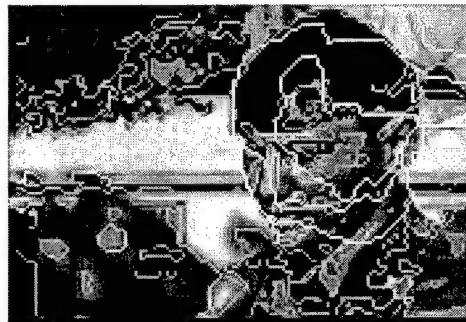


Figure 13. Video frame segmented at sub-object level

5.2.2 Special Object Indexing and Retrieval

While segmentation of general objects is the first step toward object-based indexing, it is a very tough task. This general framework may produce meaningful objects as index values. It may also only produce those low level image regions, which do not necessarily have any semantic meanings. Though these regions and their features can be used as indexing points, it is very hard for users to use it due to its lack of semantics. One way to overcome this problem is to focus on specific objects that are easy to detect and recognize, and thus useful for practical purposes. We have identified and worked on two types of such specific types of video objects: human face and regions of military interest.

5.2.2.1 Human Face Identification and Recognition

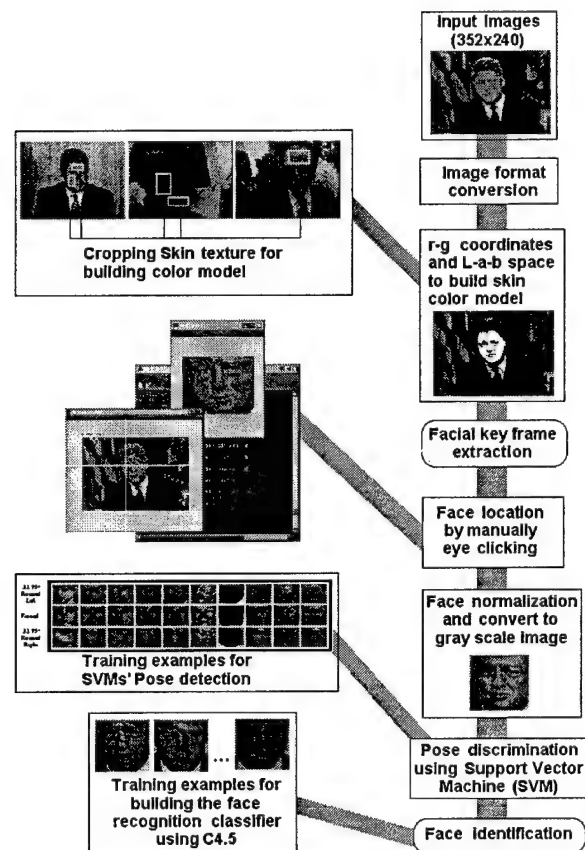


Figure 14. Methodology for Face Recognition from Digital Video Database

Human face detection and recognition can be implemented at many different scenarios. Face recognition based on still images is one such case. In this case, pictures of human face are taken from several different well-designed angles with specified pose and features are extracted for each picture. These pictures and their features are stored in a database. Query face pictures are compared against these pictures (their features in fact) to recognize these query features. This technology has matured and a commercial product is on the market.

Identification and recognition of human face from video also has many scenarios. Identification and recognition of human face from a frame-based digital video database as described in Section 5.1 is one scenario and will be described in more details in this section. Identification and recognition of human face from live video is another scenario, which we will briefly touch in Section 5.2.3.

Identification and recognition of human face from a frame-based digital video database includes two tasks: detection of face from video keyframes in a DB and recognition of face from video keyframes in a DB. Our approach is first to build skin color model from the keyframes in the database. Based on this model, we extract all keyframes that have human faces from the database. We then perform normalization to make the faces in those keyframes comparable with each other and the query face image. We then train the system with different models and perform pose discrimination using Support Vector Machine (SVM) against those keyframes that have human faces in it. We then use C4.5 to build face recognition classifier with training examples. The details are shown in the Figure above.

5.2.2.2 Regions of Military Interest

Sometimes we are interested in a class of objects from a video. For example, Unmanned Aerial Vehicle (UAV) takes video with each mission flying over 12 hours. The amount of video data is huge. We need to overcome limitations that comes with enormous amount of information and data volume, perishable contents, and that have continuous image input. The surveillance camera often operates uninterruptedly generating huge amount of image data. Not only does the vast information saturate communications bandwidth, but it also makes the human observers and analysts wear out quickly. In the area of surveillance monitoring, there is a need to identify just those regions of interest based on the image content and to screen out those redundant images containing unneeded information. The objectives for video data processing are to find regions of military interest in the shortest time period.

Obviously what constitutes a 'region of interest' will vary with the users' goals. For military usage, regions of interest are generally regions that have suspicious weapons objects, strategic highways, and buildings, or missiles in front of clouds. These targets are different from such natural scenes or objects as clouds, grass, trees, and mountains. They typically contain artificial structures such as lines, straight line, parallel lines, and curves. Thus the precise location and detection of the presence of these artificial structures will signal regions of interest. Automatically locating and detecting regions of interest from video images will help to quickly retrieve the desired scenes and will also decrease the overall duration of processing.

The architecture we proposed can respond to the query of whether artificial structures and suspected combat vehicles are detected. The architecture for ground detection takes advantage of the image understanding paradigm and it involves different methods to locate and identify the artificial object rather than nature background such as tree, grass, and cloud. Edge detection, morphological transformation, line and parallel line detection using Hough transform applied on key frame images at video shot level are introduced in our detection module. This function can also help rapidly filter incoming video and extract only those video sequences of potential interest under real time combating environment. Experimental results on video sequences acquired by Predator prove the feasibility of our approach. The algorithm is shown in Fig. 8 along with a set of sample images, each matched to the corresponding procedure. The figure demonstrates the feasibility of our approach.

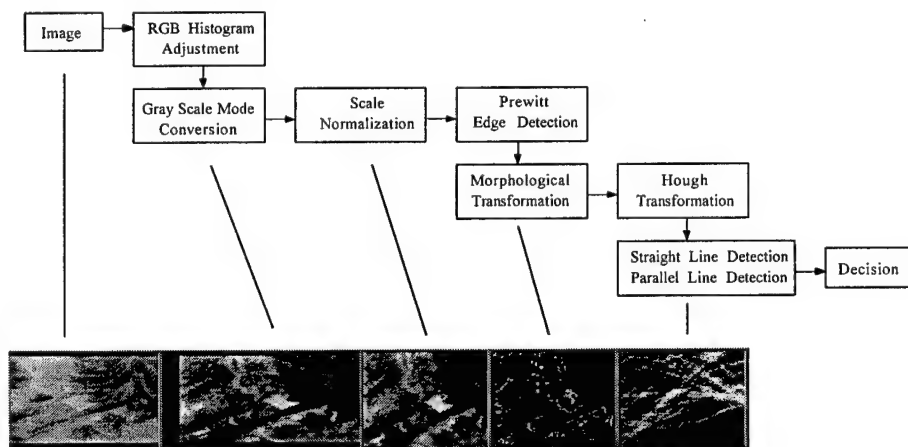


Figure 15. Algorithm for Detecting Regions of Interest

5.2.3 Object Tracking

While detection and recognition of objects are of primary importance to the understanding and indexing of video contents, tracking objects over time line is of no less significance toward video understanding. Tracking will keep objects at our fingertips and make easy access to those objects. Tracking objects will facilitate the detection and recognition because detection and recognition take time and may need to reference to the objects under investigation more than once. Tracking objects provide such opportunities.

More important is that tracking objects will provide trajectories of the objects, and their behaviors. These trajectories and behaviors can be used to further describe objects for indexing and understanding. Tracking will also identify events. Events are another type of video contents that humans often go after.

We have studied the state-of-art in this area. Statistic robust estimation technique is one of the best technologies that have been applied for this purpose. We followed this approach and experimented on some of Predator video data. It shows excellent results. However, it executes expensive calculation and is hard to implement in real-time without special equipment.

Finding optimal destination is the approach the traditional Robust Estimation has been following for the solution of optimization problem. It obtains a robust estimation by maximizing the probability of the observed image sequence with respect to the unknown motion. This approach focuses on the destination, without considering the process to arrive at the destination. Following an optimal process is another way to solve the problem, which is what Genetic Algorithms (GAs) is trying to achieve.

5.2.3.1 Object Tracking based on Genetic Algorithm

The process of natural selection leads to evolution as a result of adaptive strategies being continuously tested for their fitness as it is the case for closed-loop control. Reasoning by analogy, one attempts then to emulate computationally the 'survival of the fittest' for complex and difficult problems as those encountered in detection and homing. Evolutionary Computation (EC) in general, and GAs in particular, mimic what nature has done all along and it does that by using similar principles. GAs are further defined when one provides a specific strategy for choosing the offsprings and/or the next generation. Simulated breeding is one of the possible strategies where offsprings are selected according to their fitness. Note also that simulated breeding is conceptually similar to stochastic search in general, and to simulated annealing in particular, for the case when the size of the offspring population is limited to one individual only.

GAs, as examples of evolutionary computation, are non-deterministic methods, similar to stochastic approximations, that employ cross over and mutation as selection strategies for behavioral optimization and adaptation. GAs work by maintaining a constant-sized population of candidate solutions known as individuals. The power of a genetic algorithm lies in its ability to exploit, in a highly efficient manner, information about a large number of individuals. The search underlying GAs is such that breadth and depth - exploration and exploitation - are balanced according to the observed performance of the individuals evolved so far. By allocating more reproductive occurrences to above average individual solutions, the overall effect is to increase the population's average fitness.

The optimization seeks to improve performance toward some optimal point or points. In the other word one seeks improvement to approach some optimal point. It implies two performance of (i) the speed of convergence, and (ii) the precision of the optimal points converged. GAs are different from normal optimization and search procedures in four ways:

- (i) GAs work with coding of parameter set, not the parameter themselves
- (ii) GAs search from a popular of points, not a single point
- (iii) GAs use payoff (objective function) information, not derivatives knowledge
- (iv) GAs use probabilistic transition rule, not deterministic rules.

We have implemented Automated Multiple Motion Analysis (AMMA) system to us GA to solve the enhancement and detection of moving objects while the background is complex and moving. AMMA system architecture consists of three main components. First, the optical flow is to estimate the spatial and temporal derivatives using local window (kernel) between two continuous frames. The motion analysis model defines the constrain function to estimate the coherent surface motions using local flow information derived from former component. Genetic Algorithms (GAs) then search for the optimal surface motion and the corresponding locations based on the fitness generated from those constrain function. The diagram of system architecture is shown in Fig. 16, while each component is described respectively.

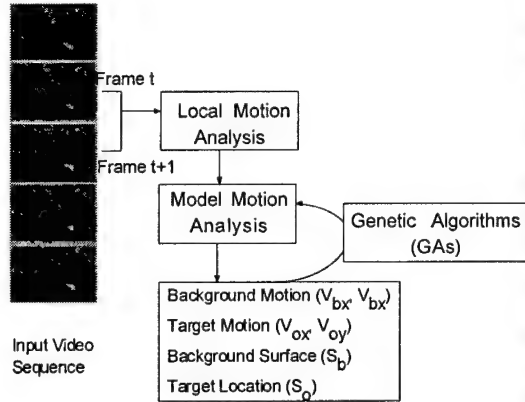


Figure 16. AMMA System Architecture

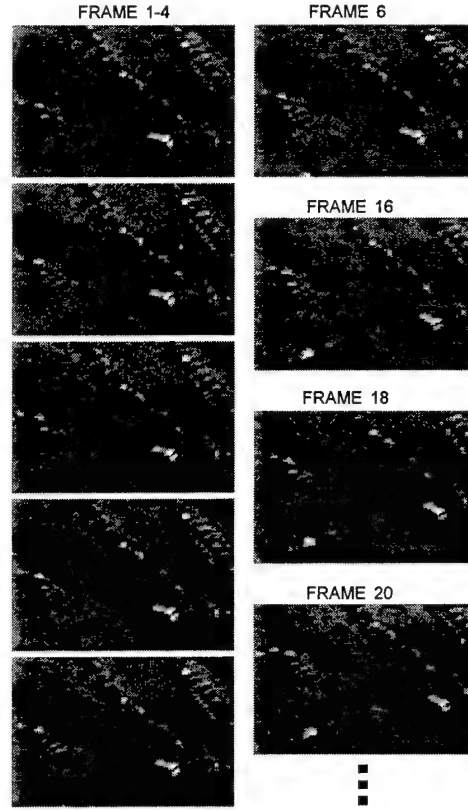


Figure 17. Example of a video sequence

We compute an array of 'flow vector', $(v_x(i,j), v_y(i,j))$ using 9×9 windows with 4 pixel shiftiness along vertical and horizontal direction of images. 784 (28×28) flow vectors corresponding to the same number of local windows ('tiles') having 4×4 interval are obtained. In order to implement the Genetic Algorithm Module for evolving the optimal motion sets, (V_{bx}, V_{by}) and (V_{ox}, V_{oy}) and their corresponding 4×4 tiles indicating if it supports either the background or target surface, we use GENESIS (GENetic Search Implementation System) with slight modification.

The first step in applying GAs for function optimization is to map the search space into a binary representation suitable for genetic search. We have defined five genes. The first gene is designed to describe a sets of the class (background or target) of which each 4×4 tile supports, and it comes out a total 784 bits to cover the entire geometrical location corresponding to the window location of local optical flows. The rest of four genes are to describe the surface motions of (V_{bx}, V_{by}) and (V_{ox}, V_{oy}) and the float representation is used. The Genetic Algorithms module is also implemented to consider a fitness function that satisfies the constrain derived from motion analysis model. The fitness function is defined in the following equatin and is minimized by GAs operation.

$$fitness = \sum_{(i,j) \in S_b} \sqrt{(v_{bx}(i,j) - V_{bx})^2 + (v_{by}(i,j) - V_{by})^2} + \sum_{(i,j) \in S_o} \sqrt{(v_{ox}(i,j) - V_{ox})^2 + (v_{oy}(i,j) - V_{oy})^2} \quad (17)$$

For the GA setup, we use a constant population size of 50, a crossover rate 0.6 and a mutation rate 0.001. Only two continuous frames (pairwise), F_t and F_{t+1} , are considered for a GAs experiment. To speed up the GA convergence, the final outcome will be carried over to succeeding experiment for next continuous frames, F_t and F_{t+1} .

Figure 18 illustrates the result of the local flow vectors and Figure 19 indicates the more significant target motion while of background coherent motion (V_{bx} , V_{by}) = (-0.06, 0.83) is found by GA and the motion cancellation is done.

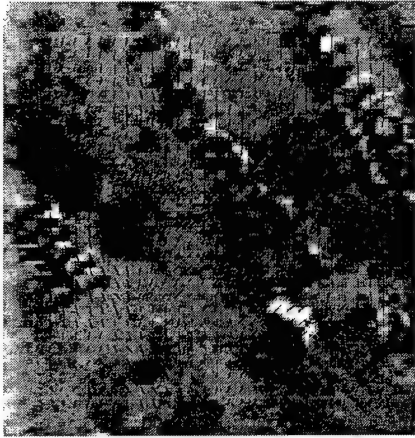


Figure 18. Local flow vector

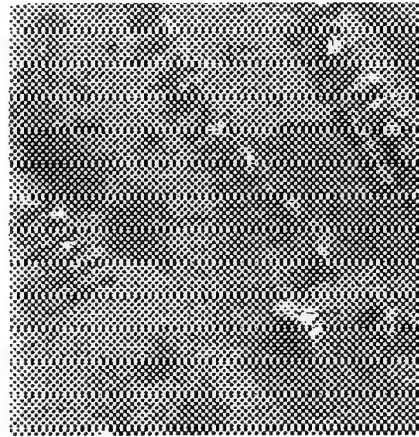


Figure 19. After cancellation



Figure 20. Target region found

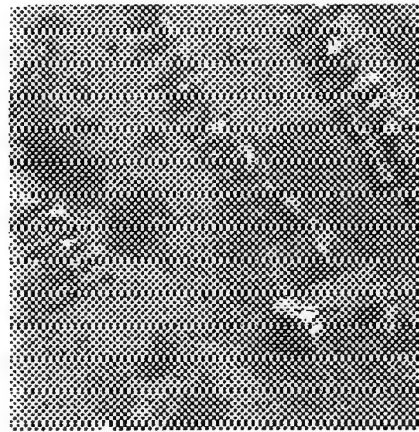


Figure 21. The final result

We can see that the moving direction of vehicle is facing the right bottom corner and also most background motion can be eliminated. Furthermore, after we apply the outcome of the first gene representing the background / target surface, from Figure 20 we observe that the target region is precisely segmented when the bit 1 indicates a target region and bit 0 masks the background. Figure 21 shows the target is popped up and the remaining false positive regions are reduced if the regions detected are weighted by the magnitude of flow vector after the background motion is cancelled.

5.2.3.2 Object Tracking based on DC and AC parameters from MPEG sequence

There have been contradictory reports about the performance of GAs algorithms. We have not been able to obtain substantially meaningful data since we only experimented on a very isolated sample. Meanwhile, we made efforts to explore the approach of using DC and AC information in MPEG sequence to implement tracking. Since most video sources are stored in the compressed form, processing in the compressed domain without full decompression is more efficient.

Video processing in compressed domain has also been an active research area in the past few years. Since most video sources are stored in compressed form, the input and output of most processing algorithms are required to be in compressed form. Algorithms implemented directly in compressed domain can offer significant computation savings when compared with approaches on the decompressed data. However, few researchers have addressed the problem of motion segmentation and tracking in the compressed domain. We proposed here an integrated computationally efficient method to achieve this task and address some important issues concerning object tracking in compressed domain.

Our overall scheme is composed of three stages: (1) extraction of reduced sized frames, specializing DC or DC+2AC, (2) feature detection and matching, and (3) feature clustering and tracking. This scheme does not require re-computation of a dense motion field, although refinement of the motion estimation at some points may be necessary at times.

DC and DC+2AC frame extraction

A DC frame is extracted by retaining only DC coefficient of each 8x8 block. the frame size is therefore reduced to 1/64 of the original size. The DC+2AC frame retains two additional AC coefficients for each block. We adopted the extraction scheme proposed by Yeo and Liu. Both DC sequence and DC+2AC sequence are tested with feature detection. It turns out that DC sequence does not have enough features to be presented to the following stages.

Feature detection and matching

We track objects using detection and tracking of object features, which can be strong edges or corners. (Currently, only corners are detected and matched.) Not all motion vectors can be conveniently employed for segmentation and tracking, because motion estimation is often not stable when insufficient spatial features are available to overcome the aperture problem. On the other hand, feature points have provided most information to identify the objects.

Our feature detection and matching algorithm is similar to the method proposed by Smith and Brady. Each pixel in the image is associated with a local region possessing similar brightness. Non-linear filtering is employed to find this region. The feature detector is then based on minimization of the region. The matching criteria for the feature points are the brightness and the x and y components of the USAN center of gravity (defined by Smith and Brady). The motion vectors were extracted directly from the MPEG sequence. They are often not accurate at the locations of the feature points because only one motion vector was encoded for every macro-block. Therefore, the feature points need to be re-matched between adjacent frames in order to improve the motion estimation accuracy. However, by using the motion information of the macro-block where the feature point resides, the searching area can be reduced.

Feature based motion segmentation and cluster tracking

Once the features are detected and matched between adjacent frames, they will be segmented into clusters based on affine motion model, described by six parameters. The segmentation algorithm is a modified K-mean approach. Each cluster is started with a two-parameter motion model, that is, constant optical flow. This is replaced with the affine model once enough vectors have been included into the cluster. The estimation of the parameters are achieved by least-square fit. The distance function used for comparing a candidate motion vector μ with the model prediction μ_m is calculated by,

$$D = \frac{\frac{\|\vec{\mu} - \vec{\mu}_m\|}{\|\vec{\mu}\| + \|\vec{\mu}_m\|}}{2} + \sigma \quad (18)$$

where σ is of the order of the error estimate of the motion vectors. Once a list of independent clusters has been found for each frame, each cluster will be characterized by its bounding box, centroid, and motion

model estimated. The cluster list will be maintained for N frames. Each cluster will try to match over these N frames to obtain robust tracking.

The attributes used for tracking are the motion model and the shape of the cluster. A radio map will be employed to represent the object shape. The radio map is a set of distance from the centroid of the cluster to the boundary vertices, calculated at equal angle increments. The advantage of radio map representation over the simple convex hull representation lies in the fact that it reduces influence of occasional erroneous boundary points which may dramatically change the object shape in the convex hull representation.

Test results

We have tested the approach on three video sequences. One frame from each of the original sequences is shown in Fig 22. The extracted DC and DC+2AC frames with feature detection are shown in Fig 23 and Fig 24 respectively. The DC frames have been enlarged 4 times and the DC+2AC frames 2 times for ease of visual comparison. While the original frame is in color, we use only luminance intensity for the extracted frames and for tracking. The features detected are corners in the current implementation. Edge detection may also be incorporated to improve the tracking performance. It is noted that there are very few feature points in the DC frames, which may lead to inaccurate or unstable clustering. More details can be identified in DC+2AC frames, but the computation is increased. Therefore, we will use DC+2AC sequence to perform object segmentation and tracking.

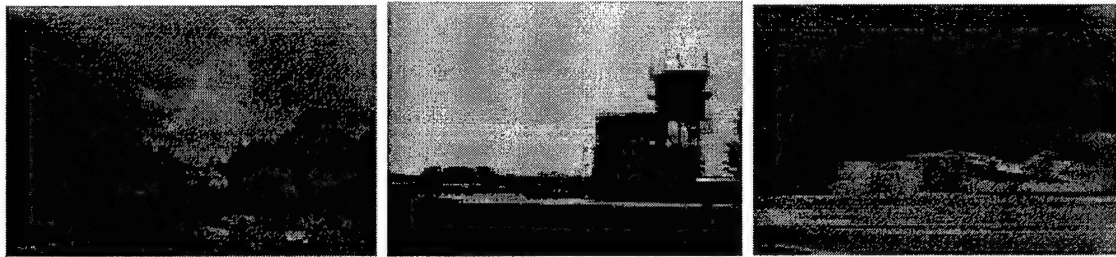


Figure 22. Original Frames

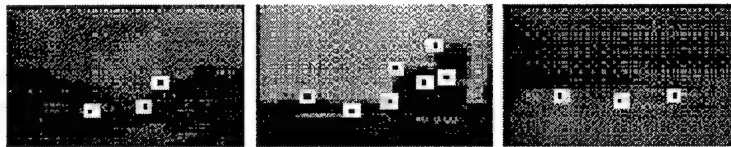


Figure 23. Extracted DC frames

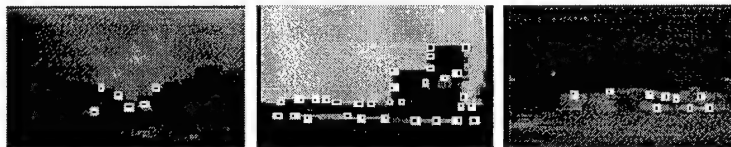


Figure 24. Extracted DC+2AC frames

5.2.3.3 Object Tracking for Face Detection

The framework for General Object Indexing and Retrieval introduced in Section 5.2.1 can also find application for tracking human face, thus helping human face detection. We use motion information for

finding face region and face detection using Decision Trees (DT) in $r-g-b$ and $L-a-b$ color spaces. Motion information is extracted from the consecutive frames adjacent to the key frames and then the phases of color segmentation in normalized $r-g-b$ and $L-a-b$ spaces are performed. The decision tree that has been trained to classify the facial pixels is then used for locating the face region based on the color features. The architecture in Fig25 shows the combination of motion extraction information and color segmentation for face detection.

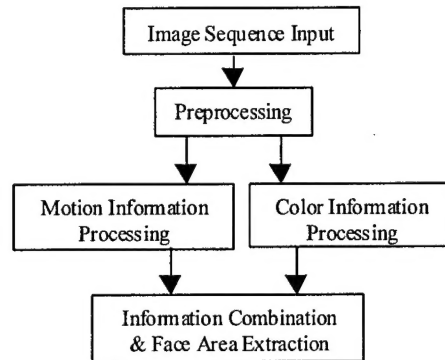


Figure 25. Face Detection System Architecture

Two video sequences shown in Fig 26 are used to test our face detection algorithms. The errormaps shown in Fig. 27 are produced using Eq. (6). The search region for face detection can be constrained and reduced to smaller area after the x-y axis projection is applied. The outcomes of motion extraction for the keyframes are shown in Fig.28 and then passed to next color segmentation module for locating the exact face region.

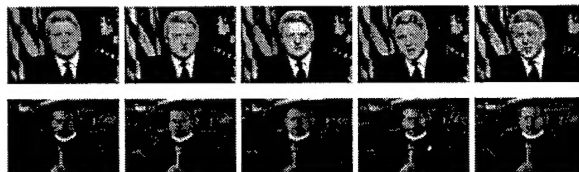


Figure 26. Sample frames of two video sequences

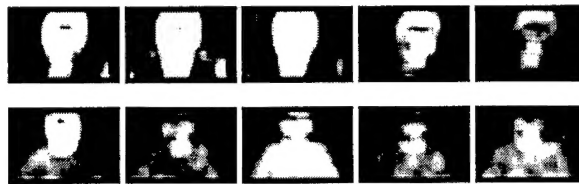


Figure 27. Errormaps corresponding to video sequences in Figure 26

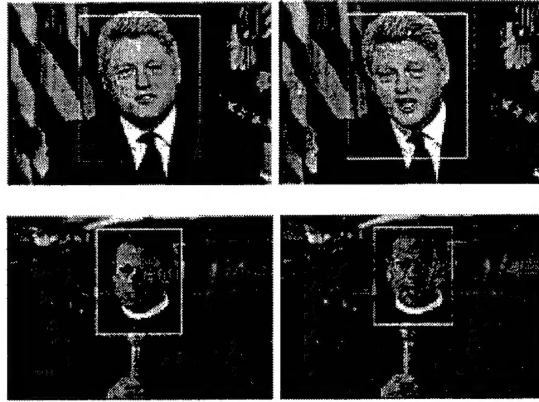


Figure 28. Rough face region of keyframes found by motion detection

Using the color face model combining the six color features of r , g , b , L , a , and b , face candidates can be segmented using decision trees. In order to generate DT color model on r - g - b space and the second color scheme based on L - a - b space, we randomly draw several images among the video database and manually crop their face region as our training data. The trained decision trees are then used to test the each pixel of face regions found by optical flow.

After applying DT trained for r - g - b and L - a - b color models, we can box the face region. It is unavoidable that some non-face regions are introduced, therefore the post-processing is needed. We apply x - y axis projection again to group and box these segmentation regions and receive precise results. Fig.6 illustrates the final results of face detection. The found face boxes are then passed for further process of either face recognition module or video indexing and query module.

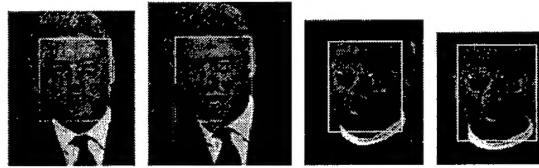


Figure 29. Final results (boxed) of face detection using color model built by DT

5.2.4 Object Recognition

No matter how we achieve the objectives of segmenting objects from video, there is always an essential problem we need to deal with: associating semantics to the objects segmented. There are many approaches to this problem, including:

- Automatic target recognition (ATR)
- Subject-based semantics as developed at Princeton University
- Learning semantics from user, exploit domain specific semantics, and
- Using visual semantic templates

Automatic target recognition is a large topic. We did not spend efforts on it specifically, though we provided some data for students at George Mason University and co-authored a paper with them on recognizing battlefield tanks. The paper describes the architectures for pattern classification using machine learning (ML). AQ18 and Decision Tree (DT) are used for learning the 2D signal introduced from SAR images consisting of three classes of combat vehicles – BMP-2, BTR-70, and T-72 tank from MSTAR SAR imagery database. Principle Component Analysis (PCA) method and whitening transformation are first used to reduce the dimensionality of input vector from 465 features, cropped from SAR images, to 30 features. We report three experimental results – DT with and without using PCA for input vectors, and AQ18 to learn the input features with PCA. We use 67 images drawn from database

with similar aspect ($\pm 15^\circ$) for training while unseen 48 images are used for testing. The method has been demonstrated the high accuracy of performance in terms of faster learning and recognition rates using AQ compared with Decision Tree after PCA and whitening transformation.

6. IMPORTANT FINDINGS AND CONCLUSIONS

The work performed has focused on the research and development of technologies for the indexing and retrieval of digital video contents. The methodologies are built upon the theories of frame-based access and object-access, which allows users to index and retrieve video contents accordingly. Research was also performed on access at levels of video scene and video programs. For these purposes, algorithms for video shot boundary detection and keyframe extraction were developed. Closed captioning is extracted and associated with video shots so that keyword searching can be performed to retrieve video shots. Different approaches are used to segment video objects, including motion-based segmentation, the combination of motion information and image cues for segmentation. Special objects such as human face and military target of interest were specifically studied as indexing points. Techniques for object identification and tracking were also studied in terms of indexing and retrieving video contents. A comprehensive prototype has been built and software developed for frame-based access technologies

7. SIGNIFICANT DEVELOPMENTS

The following table lists the software developed during this project.

Software	Descriptions	Tools
Shot boundary detection and keyframe extraction from MPEG II	Detects significant changes in frames to identify shots. Then selects key frames from shots to represent the contents of each shot. Works for MPEG II	Unix C
Shot boundary detection and keyframe extraction from MPEG I	Detects significant changes in frames to identify shots. Then select key frames from shots to represent the contents of each shot. Works for MPEG I and MPEG II.	Unix C
Digital Video Database System	Integrate IBM QBIC, keyframes, closed captioning, text search based on keywords, Netscape plug-ins	MS Visual C, Perl, Unix C
Face Detection	Perform face detection on keyframe and living video	Matlab Unix C
Region of Interests (line detection)	Line detection using Hough Transform performed in Predator Keyframe	Matlab
Optical Flow	Optical flow program	Visual C++
Target Detection and Tracking	Detect ground vehicle using optical flow and Genetic Algorithms (GAs)	Matlab Unix C
Image Segmentation	Object-based video processing using local color, texture, flow information	Matlab
Demo Video Clips	MPEG Demo	Xing or other MPEG player

8. IMPLICATIONS FOR FUTURE RESEARCH

Our experience indicates that our technologies for frame-based digital video indexing and retrieval have matured. The challenge is to further development it into a comprehensive commercial product. In addition, our experience shows that object-based indexing and retrieval has great advantages over frame-based methodologies. It is poised as the future direction and the related technologies have wide applications. We, together with other researchers in the field, have made progress, however there are still great strides to be made.

Algorithms for motion based object segmentation have been under extensive research and many excellent algorithms have been developed. However, a satisfactory approach toward real-time implementation is still far from reach. Combining motion and other image cues to implement object segmentation will make segmentation more accurate, but add more requirements to the consumption of computing power. Another extremely challenging task is to identify semantics for segmented objects. Object-based indexing and retrieval will not have practical application until these problems are solved.